03

Para saber más: filters

Filter es una de las características que componen la especificación Servlets, que estandariza el manejo de solicitudes y respuestas en aplicaciones web en Java. Es decir, dicha función no es específica de Spring y, por lo tanto, puede usarse en cualquier aplicación Java.

Es una característica muy útil para aislar códigos de infraestructura de la aplicación, como por ejemplo, seguridad, logs y auditoría, para que dichos códigos no se dupliquen y se mezclen con códigos relacionados con las reglas comerciales de la aplicación.

Para crear un Filter, simplemente cree una clase e implemente la interfaz `Filter` en ella (paquete jakarta.servlet). Por ejemplo:

```
@WebFilter(urlPatterns = "/api/**")
public class LogFilter implements Filter {

    @Override
    public void doFilter(ServletRequest servletRequest, Servle
        System.out.println("Requisição recebida em: " + LocalD
        filterChain.doFilter(servletRequest, servletResponse);
    }

}
```

COPIA EL CÓDIGO

El método `doFilter` es llamado por el servidor automáticamente, cada vez que este filter tiene que ser ejecutado, y la llamada al método

`filterChain.doFilter` indica que los siguientes filters, si hay otros, pueden ser ejecutados. La anotación `@WebFilter`, agregada a la clase, indica al servidor en qué solicitudes se debe llamar a este filter, según la URL de la solicitud.

En el curso, usaremos otra forma de implementar un filter, utilizando los recursos de Spring que facilitan su implementación.